VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Identifakce lokálních komunit v sociálních sítích

Local Community Identification in Social Networks

2013 Jan Freiherr

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Jan Freiherr**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Identifikace lokálních komunit v sociálních sítích
Local Community Identification in Social Networks

Zásady pro vypracování:

Cílem práce je provedení průzkumu existujících přístupů v oblasti detekce komunit v sociálních sítích, návrh a implementace vybraných metod a aplikačního prostředí pro experimenty.

1. Průzkum a popis existujících přístupů.
2. Návrh a implementace vybraných metod.
3. Návrh a implementace počítačové aplikace pro provádění experimentů.
4. Návrh, realizace a hodnocení experimentů.

Seznam doporučené odborné literatury:

J. Chen, O.R. Zaiane, R. Goebel. Local Community Identification in Social Networks. CASoN 2009, Fontainebleau, Francie.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Mgr. Miloš Kudělka, Ph.D.**

Datum zadání: 18.11.2011

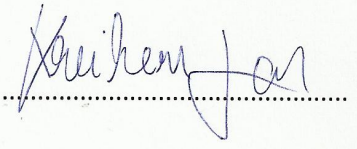
Datum odevzdání: 07.05.2013

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 4. 5. 2013



Rád bych na tomto místě poděkoval všem, zvláště pak mému vedoucímu bakalářské práce Mgr. Miloši Kudělkovi, Ph.D., protože bez nich by tato práce nevznikla.

Abstrakt

Cílem této bakalářské práce bylo prozkoumat určitou oblast sociálních sítí, a to strukturu komunit. Konkrétně studiím vyhledávání lokálních a globálních komunit v sítích se v dnešní době věnuje velká pozornost. V této práci jsem představil několik vybraných metod pro vyhledávání lokálních komunit. Na těchto metodách jsem provedl několik experimentů na reálných sítích a na základě dosažených výsledků jsem porovnal jejich přesnost. V poslední části této práce jsem předvedl aplikaci pro vizualizaci výsledků jednotlivých experimentů.

Klíčová slova: graf, síť, sociální síť, vyhledávání komunit

Abstract

The aim of my bachelor thesis is to describe one of the most obvious features of social networks, community structure. There has been much research on the subject of local and global community structure identification. This thesis explores several local community identification methods in order to compare them based on set of experiments on real-world networks. According to the experimental results of these tests I have proved difference in accuracy of various methods. In the final part of this thesis I have presented my implementation of application for displaying results of these experiments.

Keywords: graph, network, social network, local community identification

Obsah

Seznam tabulek

Seznam obrázků

Seznam výpisů zdrojového kódu

1 Úvod

Žijeme v době, kdy mezi nejcennější věci vůbec nepatří nic jiného než informace. Není proto divu, že se oboru analýzy dat a informací z nich získaných věnuje tolik pozornosti. V mnoha případech lze určité datové soubory reprezentovat pomocí sítí. Mezi typické příklady takových sítí patří jakékoliv organizované struktury, World Wide Web (WWW), autorská spolupráce na vědeckých publikacích, sociální sítě (Facebook, Twitter), biologické sítě (nervová síť) a mnoho dalších.

Sítě mohou být reprezentovány pomocí vrcholů a hran, kdy jednotlivé vrcholy představují členy sítě a hrany reprezentují jejich vzájemnou interakci. Takové hrany mohou v sociálních sítích představovat jakoukoliv komunikaci (email, telefonní hovory), přátelství nebo spolupráci. Ze studie sítí se stal mocný nástroj pro pochopení právě těchto vazeb mezi jednotlivými členy. Pomocí vyhodnocování všech těchto interakcí můžeme v sítích identifikovat komunity.

Komunita v síti může být popsána jako skupina vrcholů, ve které je hustota hran mezi jejími členy navzájem větší než hustota hran spojující její členy se zbytkem sítě. V sociálních sítích mají uživatelé patřící do stejné komunity většinou nějakou společnou charakteristiku, na základě které můžou být odlišováni od ostatních členů sítě. Postupem času sociální sítě přerostly do takových rozměrů, že je nemyslitelné vyhledávat komunity manuálně. Díky této skutečnosti vznikla potřeba vyvinout algoritmy pro vyhledávání komunity v sítích.

Problémem vyhledávání komunit v sociálních sítích se společnost zabývá už mnoho let. V minulost byly představeny algoritmy, které ale většinou požadovaly jako vstupní data kompletní znalost celé sítě. Takové algoritmy nazýváme *globální*. Tento přístup k problému je z praktického hlediska v dnešní době použitelný pouze pro ne příliš rozsáhlé sítě. Získat totiž globální informace o struktuře celé sítě nemusí být vždy jednoduché. Můžeme narazit na sítě příliš rozsáhlé nebo příliš dynamicky se rozvíjející. Jako takovou síť můžeme považovat např. WWW (World Wide Web). U takového druhu sítě je nemyslitelné získat její kompletní strukturu, která by byla následně využita jako vstupní informace pro tyto algoritmy. V takových případech, kdy nelze použít globální metody, je nutné využít algoritmů pro vyhledávání *lokálních* komunit.

Základním rozdílem mezi globální a lokální metodou je jejich cíl. Cílem globálních metod je najít způsob, jak rozdělit síť takovým zbůsobem, že nalezené komunity budou pokrývat všechny vrcholy. Lokální metody volí jiný přístup. Algoritmy pro vyhledávání lokálních komunit hledají ideální komunitu v okolí určitého vrcholu, který je pro ně požadovaný jako vstupní parametr. Druhým důležitým rozdílem je dostupnost informací. Globální metody mají přístup k informacím o struktuře celého grafu v kterékoliv fázi algoritmu. Na rozdíl od globálních metod jsou metodám lokálním dostupné pouze informace o okolních vrcholech. Většina lokálních metod neprohledá postupně celou síť, ale zastaví se po dosažení určitého kritéria. Většinou se jedná o nějakou podmínku, která se liší v závislosti na použitém algoritmu. Nejčastěji to je předem definovaný způsob výpočtu nějakého koeficientu kvality pro výslednou komunitu.

V této práci se budu věnovat druhému jmenovanému typu - algoritmu pro vyhledávání *lokálních* komunit. Práce je koncipována následovně. V první kapitole se budu věnovat několika metodám pro vyhledávání lokálních komunit. Popíši, jak fungují a u některých z nich předvedu na příkladech jejich nedostatky. Poslední metodě (Metrika L) budu v této kapitole věnovat větší pozornost než ostatním metodám. Tuto metodu jsem naimplementoval. Ve druhé kapitole se budu věnovat několika experimentům prováděných především na zmíněné naimplementované metodě. Pokusím se vyhodnotit identifikované komunity a jejich kvalitu. V poslední kapitole popíši aplikaci, kterou jsem navrhl a naimplementoval.

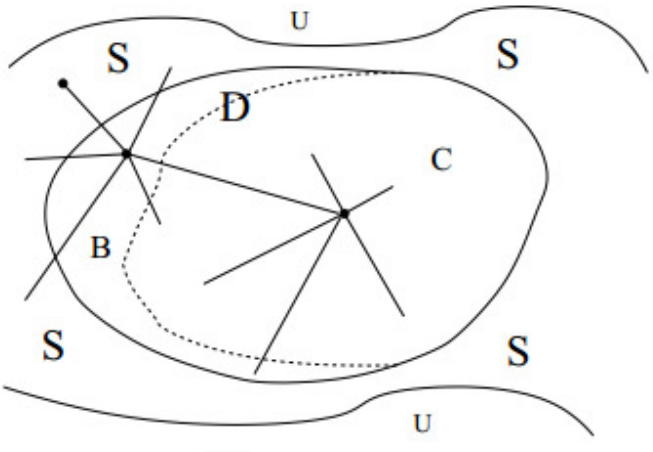
2 Algoritmy pro vyhledávání lokálních komunit v sítích

Jak už bylo zmíněno v úvodu, komunita je skupina vrcholů v grafu, pro kterou platí, že hustota hran mezi jejími členy navzájem je větší než hustota hran mezi jejími členy a zbytkem grafu. K identifikaci lokálních komunit není třeba znát strukturu celého grafu, ale stačí znát pouze lokální informace o jeho části. Pojmem lokální informace rozumíme skupinu vrcholů, u kterých známe všechny jejich sousedy. Jediný způsob, jak tuto skupinu vrcholů rozšířit, je navštívit jejich sousední vrcholy a získat seznam jejich sousedů. Toto je základní princip, jak algoritmy pro vyhledávání lokálních komunit získávají informace, na základě kterých identifikují komunity. Jednotlivé algoritmy se od sebe vzájemně liší tím, jak tyto komunity vyhodnocují a na základě podmínek, podle kterých do vyhledávaných komunit přidávají nové členy.

V této kapitole se budu věnovat průzkumu a popisu existujících metod pro vyhledávání lokálních komunit v sítích.

2.1 Zavedení značení podgrafů

Mějme síť reprezentovanou grafem G. Algoritmy pro identifikaci lokálních komunit vyžadují jako vstupní údaj dokonalou znalost části grafu. Tato část může být reprezentována skupinou vrcholů nebo samostatným vrcholem. Označme tuto část grafu G jako podgraf D. Pokud existuje tento podgraf, tak máme limitované informace o jeho sousedních vrcholech, které ale nepatří do D. Všechny tyto sousední vrcholy patří do podgrafu, který nazveme S. Výraz limitované informace chápeme tak, že neznáme všechny sousedy vrcholů v S. Jediným způsobem, jak získat další informace o grafu G je navštívit tyto vrcholy a získat seznam jejich sousedů. Tento postup se využívá při vyhledávání vhodných členů pro hledanou lokální komunitu.



Obrázek 1: Ukázka značení grafu

Tento postup probíhá následovně. Vrchol $s \in S$ je odebrán z S a přidán do D. Nyní můžeme všechny sousedy tohoto vrcholu přidat do S a následně od nich získat seznam, obsahující informace o dalších vrcholech grafu G. Při tomto postupu dochází k tzv. přidávání jednoho vrcholu v každém kroku algoritmu. Tímto způsobem prozkoumávají algoritmy pro vyhledávání komunit síť tak dlouho, dokud nevyhodnotí, že už neexistují další vhodné vrcholy pro přidání do výsledné komunity.

Dále existují dva podgrafy množiny D. Jedná se o okrajový (periferní) podgraf B, který obsahuje jen ty vrcholy, pro které platí, že minimálně jeden z jejich sousedů nesmí být členem D. Druhý podgraf monžiny D je označen jako C - centrální. Z vrcholů množiny C vedou pouze takové hrany, které mají dva koncové body v D. Zbytek sítě, o které nemáme žádné informace, označíme jako množinu U.

Tento nebo obdobný způsob označení podgrafů při vyhledávání lokálních komunit byl použit také v [1, 3]. Nicméně samotné metody se od sebe vzájemně liší.

2.2 Lokální modularita - R

Jako první si představíme metodu, kterou ve svém článku popsal A. Clauset [1]. Tato metoda vyhodnocuje kvalitu komunity tak, že se soustředí na tzv. ostrost (sharpness) okrajové části komunity (podgrafu B). Ostrost je nezávislá na velikosti komunity. Pokud můžeme říct, že komunita má *"ostrou"* okrajovou část, pak taková komunita bude obsahovat pouze několik hran, které vedou do množiny S, a zároveň bude mít mnoho hran, spojujících tuto okrajovou část komunity s její centrální částí (C).

2.2.1 Výpočet

Vzorec pro výpočet *lokální modularity - R* je definován následovně:

$$R = \frac{B_{in\_edge}}{B_{out\_edge} + B_{in\_edge}} \tag{1}$$

kde B_{in_edge} představuje počet hran, které nemají ani jeden koncový bod v množině vrcholů S (hrany vedoucí z okrajové do centrální části komunity nebo mají oba koncové body v C). Obdobně představuje B_{out_edge} ty hrany, které mají jeden konec v okrajové části komunity a druhý konec v přilehlé množině vrcholů S.

Je zřejmé, že lokální modularita bude nabývat následujících hodnot: $0 < R < 1$. Existuje jeden případ, kdy zmíněný interval není pravdivý. Taková situace může nastat, pokud při postupném získávání informací o grafu (přidávání jednoho vrcholu za druhým do D) zahrneme v této lokální časti kompletní síť. Za takových podmínek můžeme prohlásit, že R není definováno. Jako jiné řešení této situace můžeme prohlásit, že celá síť představuje velmi silně propojenou komunitu a nastavit explicitně R na hodnotu 1.

2.2.2 Algoritmus

Jako vstup, tento algoritmus vyžaduje jeden povinný a jeden volitelný parametr. Povinným parametrem je počáteční vrchol n_0, ze kterého se začnou prozkoumávat okolní vrcholy. Volitelný parametr je číslo *k*, které označuje cílový počet vrcholů. Pokud vyhledávaná komunita dosáhne tohoto počtu vrcholů, algoritmus to vyhodnotí jako splnění cílové podmínky, ukončí vyhledávání dalších vhodných vrcholů a vrátí nalezenou komunitu a její lokální modularitu. Pokud není parametr *k* zadán, bude vyhledávání pokračovat dokud tento algoritmus nezíská informace o kompletní souvislé části grafu, ve které se nachází vrchol n_0 (viz Algoritmus 1).

Existuje také varianta, kdy algoritmus bude pokračovat v přidávání nových vhodných vrcholů do *D* do té doby, než bude platit pro všechny členy sousedního podgrafu *S*, že $\Delta R < 0$. Předpokládejme, že probíhá vyhodnocování vrcholu n_i, zda je vhodný pro právě objevovanou komunitu. K tomuto vyhodnocení slouží následující vzorec:

$$\Delta R = R' - R \tag{2}$$

kde R' představuje hodnotu lokální modularity pro komunitu po přidání vrcholu n_i a R reprezentuje aktuální hodnotu lokální modularity pro tuto komunitu a musí být po každém přidání vrcholu *n* do komunity znovu vypočteno.

Algoritmus 1: Identifikace lokální komunity pomocí lokální modularity

Vstup : Síť *G* a počáteční vrchol n_0
Výstup: Lokální komunita s lokální modularitou *R*

```
Přidat n_0 do D a B;
Přidat všechny sousedy vrcholu n_0 do S ;
while |D| < k do
    foreach n_i ∈ S do
        Vypočítat ΔR;
    end
    Najít n_i s maximální ΔR;
    Přidat n_i do D;
    Aktualizovat B, S, C, R;
end
Vrátí nalezenou lokální komunitu s její lokální modularitou
```

Na podobném principu stojí základ většiny algoritmů pro vyhledávání lokálních komunit. U jednotlivých metod se liší způsob vyhodnocování, zda je daný vrchol vhodný pro přidání do komunity. Některé algoritmy pro dosažení přesnějších výsledných komunit, které obsahují méně nesprávně vyhodnocených vrcholů, používají navíc další fázi. Při této fázi se snaží vyhledat takové vrcholy, které byly v první fázi nesprávně zvolené

jako vhodné. Jedna z metod, která tento postup používá je přesněji popsána v kapitole 2.4.

2.3 Modularita - M

Upravenou verzi metody z předchozí kapitoly (viz 2.2) představil F. Luo a spol. [2]. Nazval ji *modularita M* pro vyhodnocování lokálních komunit.

2.3.1 Výpočet

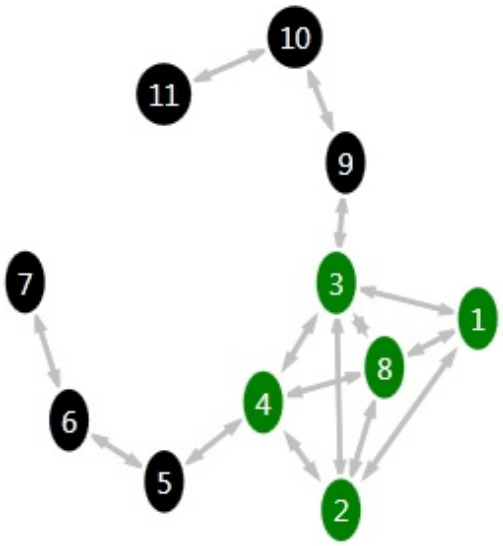
Pokud použijeme stejné značení jako v 2.2.1, bude vzorec pro posouzení kvality komunity vypadat následovně:

$$M = \frac{B_{in\_edge}}{B_{out\_edge}} \tag{3}$$

Množinu vrcholů můžeme prohlásit komunitou, pokud platí $M \geq 1$, což je ekvivalentní hodnotám metody lokální modularity $R \geq 0.5$.

2.3.2 Problém

Nedostatkem u metod M a R je jejich nízká *přesnost*. Výsledné komunity u těchto metod obsahují velké množství vrcholů, které do nalezené komunity nepatří.



Obrázek 2: Síť: Problém metod R a M

Na obrázku 2 je zobrazen graf s nalezenou lokální komunitou. Vrcholy této komunity jsou obarveny zeleně. Obě uvedené metody by v této situaci postupně vyhodnotily všechny vrcholy jako členy komunity. Jako příklad se podívejme na vrchol 5. Jeho případným přidáním do komunity by se zvětšil počet vnitřních hran a zároveň by zůstal počet externích hran stejný. Na základě takové změny počtu hran v komunitě by obě metody vyhodnotily vrchol 5 jako vhodný pro přidání do komunity. Takový postup by se opakoval u všech ostatních vrcholů z tohoto grafu.

2.4 Metoda využívající průměrný stupeň vrcholů

Většina existujících metod využívá stupně vrcholů nebo počet hran tak, že pomocí nich přímo reprezentuje hodnoty, které využívá při výpočtu kvality komunity. Nicméně výsledné lokální komunity využívající tento způsob obsahují velké množství špatně vyhodnocených vrcholů, které ve skutečnosti leží mimo danou komunitu (viz 2.3.2). Metoda využívající metriky L [3] dokázala, že je vhodnější při výpočtu pracovat s průměrnými stupni vrcholů, narozdíl od předešlých metod, které využívaly absolutní hodnotu počtu hran. Na následujícím příkladu je popsán jeden z problémů těchto metod.

Příklad 2.1
Mějme množinu vrcholů N. Tato množina bude obsahovat milion hran mezi vrcholy ležícími uvnitř N a žádnou hranu vedoucí z N k vrcholu ležícímu mimo N. Bylo by chybou považovat N za silně propojenou lokální komunitu, pokud by každý vrchol v N byl spojen pouze s jedním nebo dvěma sousedy v N. ■

Metoda využívající průměrných stupňů vrcholů k výpočtu kvality komunity bude v následující kapitole detailně popsána. V kapitole 3 se pokusím pomocí testování na reálných datech dokázat, že tato metoda dosahuje kvalitnějších a přesnějších výsledků než metody popsané v kapitolách 2.2 a 2.3.

2.4.1 Výpočet

J.Chen a spol. představil novou metodu [3], která k výpočtu nevyužívá přímo počet vnitřních a vnějších hran komunity. Na rozdíl od předešlých metod vyhodnocuje kvalitu komunity pomocí průměrného interního stupně vrcholů v D:

$$L_{in} = \frac{\sum_{i \in D} IK_i}{|D|} \tag{4}$$

kde IK_i je počet hran mezi vrcholem i a vrcholy v D. $|D|$ představuje celkový počet vrcholů v komunitě.

Podobný způsob se využije při výpočtu průměrného externího stupně vrcholů. Zohledňujeme pouze hrany, které mají jen jeden konec v komunitě:

$$L_{ex} = \frac{\sum_{j \in B} EK_j}{|B|} \tag{5}$$

kde EK_j je počet hran mezi vrcholem *j* a vrcholy v *S*. Na rozdíl od výpočtu průměrného interního stupně vrcholů se u externího počítá pouze s počtem vrcholů v okrajové části komunity - $|B|$. S vrcholy z centrální části komunity nepočítáme, protože nemají žádné hrany vedoucí z komunity ven. Známe-li hodnoty L_{in} a L_{ex}, pak je naším cílem maximalizovat následující poměr:

$$L = \frac{L_{in}}{L_{ex}} \tag{6}$$

Koeficient *L* můžeme považovat za indikátor kvality komunity. Větší hodnota *L* indikuje silněji propojenou komunitu. Naopak komunity nabývající nižších hodnot *L* považujeme za komunity se slabší vazbou mezi jejími členy a naopak silnější vazbou mezi jejími členy a vrcholy mimo komunitu. Komunity s nízkou hodnotou *L* je pro tento algoritmus těžké určit, protože z obecného hlediska přestávají být považovány za komunitu a její části se rozpadají na menší.

2.4.2 Vyhodnocování lokální komunity

Tento algoritmus je rozdělen do dvou fází. V první fázi prochází jednotlivé sousedy komunity *D* a vyhledává vrcholy, které vylepší koeficient kvality komunity *L*. Tímto způsobem se komunita rozrůstá krok za krokem o jeden vrchol. Tuto fázi nazýváme *fáze objevování*. Předpokládejme, že hodnoty L'_{in} a L'_{ex} a L' korespondují s hodnotami, vypočítanými po přidání vrcholu *i* do *D*. Při výpočtu *L* může dojít ke třem základním situacím, které vedou k $L' > L$:

1. $L'_{in} > L_{in}$ a zároveň $L'_{ex} < L_{ex}$
2. $L'_{in} < L_{in}$ a zároveň $L'_{ex} < L_{ex}$
3. $L'_{in} > L_{in}$ a zároveň $L'_{ex} > L_{ex}$

Je samozřejmé, že vrcholy patřící do prvního případu patří do komunity, protože posílí L_{in} a zároveň oslabí L_{ex}. Tento případ vždy zvýší kvalitu komunity. Vrcholy, které spadají do případu č. 2 do komunity nepatří. Jsou to vrcholy, které jsou slabě propojeny jak s komunitou L_{in}, tak i se zbytkem grafu.

U vrcholů, které patří do třetího případu, nemůžeme během první fáze rozhodnout, zda budou do finální komunity patřit. Jedná se o vrcholy, které jsou silně propojeny jak s komunitou, tak i se zbytkem sítě. Existují dva typy vrcholů, které obvykle končí v tomto případě. U prvního typu se jedná o první vrchol nové komunity, která bude do naší komunity *D* přidána vrchol za vrcholem, dokud v ní neskončí celá. Druhý typ nazýváme "hub". Jedná se o vrchol, ze kterého vede velké množství hran k vrcholům v komunitě a zároveň velké množství hran vedoucích k vrcholům mimo komunitu. Takový vrchol může např. spojovat dvě komunity. Vrcholy typu *hub* ve výsledné komunitě nechceme. Každopádně během první fáze nelze rozhodnout, o který z těchto dvou typů vrcholu se jedná. Jako řešení tohoto problému zvolili tvůrci tohoto algoritmu přidání vrcholu, který patří do třetího případu, do komunity *D* během první fáze.

2.4.3 Algoritmus

Algoritmus 2: Identifikace lokální komunity pomocí metriky L

Vstup : Síť *G* a počáteční vrchol n_0
Výstup: Lokální komunita s koeficientem kvality *L*

1. fáze - Objevovaní komunity ;
Přidat n_0 do *D* a *B* ;
Přidat všechny sousedy vrcholu n_0 do *S* ;
do
  foreach $n_i \in \S$ **do**
    Vypočítat L'_i;
  end
  Najít n_i s maximální L'_i, přidat n_i do *D*, pokud patří do prvního nebo třetího případu, jinak odebrat n_i z *S*;
  Aktualizovat *B*, *S*, *C*, *L*;
while $L'_i > L$;
2. fáze - Vydnocování ;
foreach $n_i \in D$ **do**
  Vypočítat L'_i. ponechat n_i pouze pokud náleží do prvního případu;
end
3.;
if $n_0 \in D$ **then**
  Vrátí *D* ;
else
  Komunita pro n_0 neexistuje
end

Po té, co byly všechny vrcholy, které byly kvalifikovány jako vhodné pro přidání do komunity, přidány, končí první fáze. Nyní začne druhá fáze nazvaná *fáze vyhodnocování*. V této fázi vyhodnotíme každý vrchol komunity tak, že ho vyjmeme z komunity a znovu

pro něj vypočteme hodnoty kvality komunity L_{in} a L_{ex} a L. Následně dojde k opětovnému porovnání těchto hodnot se situací, při které by byl vrchol znovu přidán do komunity. Pokud při tomto porovnání dojdeme k závěru, že vrchol spadá do případu číslo jedna, můžeme finálně prohlásit, že vrchol patří do komunity D.

V druhé fázi dojde k rozlišení vrcholů dvou typů, které byly v první fázi vyhodnoceny jako třetí případ. Vrchol, u kterého se jednalo o první vrchol komunity, která byla následovně celá přidána do D, bude nyní patřit do prvního případu. Většina jeho sousedů byla totiž v první fázi přidána do komunity. Naopak hub bude stále patřit do třetího případu.

Jako finální krok toho algoritmu se vyhodnotí, zda je počáteční vrchol n_0 stále členem komunity. Pokud byl vrchol n_0 během druhé fáze z komunity odebrán, znamená to, že komunita pro daný vrchol neexistuje. Naopak, pokud je vrchol n_0 stále členem komunity D, algoritmus vrátí nalezenou lokální komunitu a její koeficient kvality L. (viz Algoritmus 1: Identifikace lokální komunity)

```
//vypocet hran z vrcholu "vertex" do komunity(Indi) a do zbytku site (Outi)
int Indi = 0, Outdi = 0; //
int DifferenceB = 0;
   foreach (int edge in G.edges[vertex])
   {
      if (G.B.Contains(edge))
      {
         Indi++;
         // vypocet noveho podgrafu B(B_) po pripadnem pridani vrcholu vertex
         boolean StillB = false; // indikuje, zda-li "vertex" ma zustat v komunite
         // prochazeni vsech sousedu vrcholu "vertex"
         // kontrola, jestli ma vrchol, ke kteremu vede hrana "edge" v B zustat
         //(maji i po pridani vrcholu "vertex" dalsi sousedy v S)
         foreach (int Bedge in G.edges[edge])
         {
            if (G.S.Contains(Bedge) && Bedge != vertex) StillB = true ;
         }
         if (! StillB ) DifferenceB++;
      }
      else Outi++;
   }
   int B_new = G.B.Count - DifferenceB + 1 ;
```

Výpis 1: Ukázka kontroly okrajové části podgrafu

Při výpočtu jednotlivých L_i' není vhodné jednotlivé vrcholy během první fáze pokaždé přidávat a naopak během druhé fáze odstraňovat z D. Po každém takovém úkonu by bylo třeba aktualizovat všechny podgrafy (B, S, C). Během aktualizace je nutné projít celou komunitu a zkontrolovat, zda všechny vrcholy splňují podmínky pro daný podgraf, do kterého náleží, popř. daný vrchol správně přiřadit. Je jasné, že provádět aktualizaci po každém odebrání/přidání vrcholu kvůli výpočtu L_i' by bylo značně neefektivní.

Proto je pro výpočet L'_i během první fáze algoritmu použit následující vzorec:

$$L_i = \frac{\frac{Ind+2*Ind_i}{|D|+1}}{\frac{Outd-Ind_i+Out_i}{|B'|}} \tag{7}$$

kde *Ind* je součet stupňů vrcholů v *D* (2*počet hran, které mají oba konce v *D*). *Outd* reprezentuje počet hran, vedoucích z komunity ven (hrany, které mají pouze jeden konec v *D*). Hodnoty *Ind* a *Outd* musí být vypočteny po každém přidání vrcholu do *D*, protože s každým novým vrcholem v *D* se tyto hodnoty mění. Ind_i a $Outd_i$ jsou počty hran z vrcholu *i* vedoucí do *D* a do zbytku sítě. *B'* je celkový počet vrcholů v okrajové části komunity po přidání vrcholu *i*. Během výpočtu *B'* je důležité projít všechny členy podgrafu *B* po přidání nového vrcholu. Může totiž dojít k situaci, při které se z některých vrcholů v okrajové části komunity stanou vrcholy centrálního podgrafu *C*. Vrchol okrajové části komunity se stane vrcholem centrální části tak, že se všechny jeho sousední vrcholy, které nebyly členy komunity, stanou její součástí. V ukázce výpisu kódu 1 lze vidět implementaci kontroly členů *B*, jestli nedošlo k výše popsané situaci, a následný výpočet finálního počtu vrcholů množiny *B*.

Podobný postup jako při výpočtu L'_i během první fáze zvolíme i během *fáze vyhodnocování*. Opět není vhodné každý vrchol odstraňovat a přidávat, ale použijeme následující vzorec. Značení jednotlivých proměnných zůstává stejné jako v první fázi algoritmu:

$$L_i = \frac{\frac{Ind-2*Ind_i}{|D|-1}}{\frac{Outd+Ind_i-Out_i}{|B'|}} \tag{8}$$

Po skončení prvních dvou fazí algoritmu může nastat taková situace, že počáteční vrchol n_0, pro který byla hledána lokální komunita, není členem *D* (byl v druhé fázi vyhodnocen jako nevhodný pro tuto komunitu). K takové situaci může dojít, pokud patří vrchol n_0 do okrajové části komunity. Pro tuto metodu je mnohem snazší najít komunitu pro vrchol z centrální části, než pro vrchol patřící do okrajové části komunity. Pro takové vrcholy se během první fáze může stát, že při přidávání vhodných vrcholů narazí tento algoritmus na jinou komunitu, kterou bude krok za krokem přidávat k n_0. S každým takovým krokem budou vycházet hodnoty L'_i pro selekci nových vrcholů vhodněji pro novou nalezenou komunitu než pro komunitu původní. Pokud taková situace nastane, tento algoritmus jí odhalí, vyhodnotí takovou nalezenou komunitu jako chybovou a rozhodne, že pro n_0 lokální komunita neexistuje. Takový problém lze vyřešit volbou více počátečních vrcholů. Takové řešení by poskytlo tomuto algoritmu větší množství počátečních informací a tím by napomohlo udržet průběh identifikace lokální komunity ve správné části grafu.

3 Testování na reálných datech

V této kapitole se budu věnovat vyhodnocování výsledků získaných při testování algoritmu využívajícího metriky L (viz kapitola 2.1) na reálných datech. Výsledné nalezené komunity budu porovnávat s výsledky jiných algoritmů na stejných datech. Aplikuji tento algoritmus na sítě, u kterých známe rozdělení jejich členů do jednotlivých komunit, a následně budu tyto reálné komunity srovnávat s komunitami nalezenými tímto algoritmem. Pokusím se dokázat, že tento algoritmus byl přínosem pro obor vyhledávání lokálních komunit tím, že dosahuje větší přesnosti při nalezení komunit. Větší přesnosti lze dosáhnout ve výsledných komunitách menším množstvím vrcholů , které do nich nepatří a zároveň zahrnutím všech vrcholů, které do těchto komunit náleží.

3.1 Vysokoškolská liga amerického fotbalu

Jako data pro první experiment jsem použil rozpis her jedné divize ligy amerického fotbalu na sezónu 2000 [6]. Tato divize je známá pod jménem "Divize 1-A". Zajímavý na těchto datech je fakt, že celá soutěž je rozdělena na jednotlivé konference, kterých je celkem 11. Každá konference obsahuje průměrně 9 týmů. Každý z těchto týmů hraje průměrně 7 utkání v konferenci, do které patří, a průměrně 4 hry s týmy z ostatních konferencí. Mezikonferenční zápasy nejsou přesně stanoveny pro každý tým stejně. Týmy, mezi kterými jsou malé geografické vzdálenosti, spolu budou hrát častěji, než týmy, mezi kterými jsou tyto vzdálenosti velké. Celkově se Divize 1-A v roce 2000 skládala ze 110 týmů rozdělených do 11 konferencí, 5 nezávislých týmů ze škol, které se nepovažují za členy kterékoliv konference a z 615 utkání. Za pomoci grafové terminologie můžeme říct, že se tato síť skládá ze 110 vrcholů rozdělených do 11 komunit, 5 vrcholů ležících mimo jakoukoliv komunitu a 615 hran.

3.1.1 Vyhledání komunit v sezóně 2000

Tuto síť jsem aplikoval jako vstup pro algoritmus využívající průměrný stupeň vrcholů. Cílem prvního experimentu bylo zjistit, zda budou komunity nalezené pomocí této metody odpovídat jednotlivým konferencím Divize 1-A. Toho jsem docílil tak, že jsem každý ze 115 vrcholů, které reprezentují jednotlivé týmy, použil jako počáteční vrchol pro tento algoritmus. Výslednou nalezenou komunitu pro daný vrchol jsem porovnal s konferencí, do které patří tým reprezentovaný tímto vrcholem. K vyhodnocení výsledku tohoto experimentu jsem použil výpočet úplnosti (Recall), přesnosti (Precision) a harmonického průměru přesnosti a úplnosti (f-measure), které se považují za základní typy měření míry relevance.

Tabulka 1 zobrazuje vypočtené hodnoty pro přesnost (sloupec *P* - Precision), úplnost (sloupec *R*-Recall) a jejich harmonický průměr (sloupec *F* - F-measure). Hodnoty v tabulkách představují průměr, vypočítaný z hodnot pro všechny týmy v příslušné konferenci. Počet týmů v každé konferenci je zobrazen ve sloupci s označnením *n*. Týmy, pro které byl během druhé fáze algoritmu odstraněn z komunity počáteční vrchol (komunita pro ně neexistuje - viz 2.4.2), jsou sečteny pro každou konferenci ve sloupci s ozna-

Divize 1-A	Sezóna 2000					Sezóna 2006				
Konference	**n**	**X**	**P**	**R**	**F**	**n**	**X**	**P**	**R**	**F**
Mountain West	8	0	0,878	0,875	0,876	9	0	0,944	1	0,963
Mid-American	13	0	0,925	1	0,961	12	1	0,923	1	0,96
Southheastern	12	0	0,924	1	0,952	12	3	1	1	1
Sun Belt	7	0	0.531	0,51	0,513	8	3	1	1	1
Western Athletic	10	4	0,648	0,34	0,603	9	4	0,6	1	0,733
Pacific Ten	10	2	0,932	0,8	0,96	10	0	1	1	1
Big Ten	11	9	0,844	1	0,907	11	9	0,729	1	0,814
Big East	8	3	0,914	1	0,945	8	5	1	1	1
Atlantic Coast	9	2	0,949	1	0,969	12	3	1	1	1
Conference USA	10	1	0,822	0,811	0,81	12	1	1	1	1
Big Twelve	12	0	0,914	0,924	0,922	12	5	1	1	1
Celkem	110	21	0,844	0,842	0,856	115	34	0,927	1	0,952

Tabulka 1: Tabulka: Porovnání sezón 2000 a 2006 pomocí algoritmu využívajícího metriky L

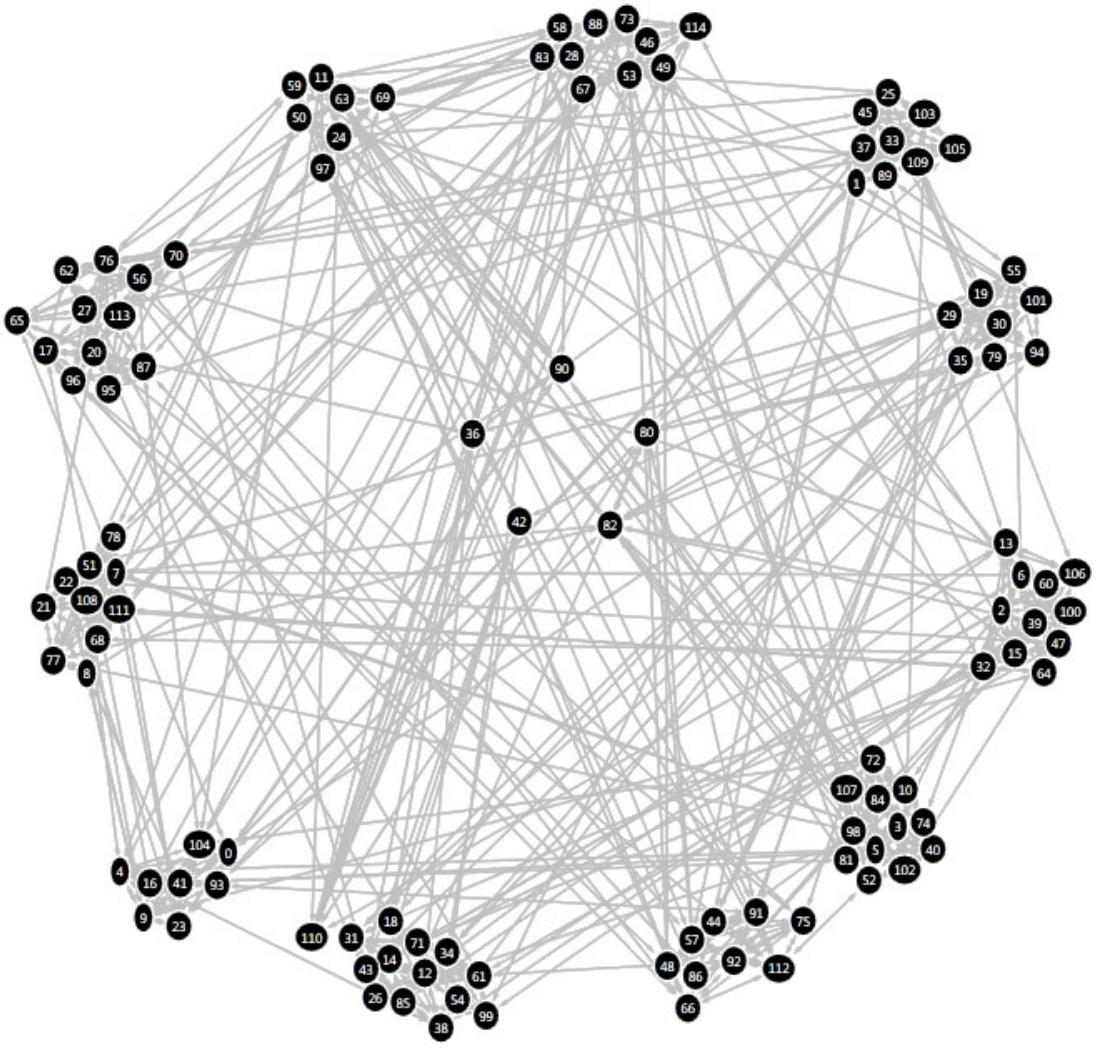
čením *X*. Je důležité si uvědomit, že se do vypočtených průměrných hodnot (*P*, *R* a *F*) nezahrnují týmy, pro které nebyla pomocí tohoto algoritmu nalezena žádná komunita.

Jako první jsem se zaměřil na komunity, nalezené mnou naimplementovaným algoritmem využívajícím metriky L. Téměř všechny týmy byly tímto algoritmem správně vyhodnoceny a přiřazeny do příslušných komunit, které složením odpovídají jednotlivým konferencím. Pro 4 z pěti nezávislých týmů byly nalezeny komunity, které většinou představovaly určitou část konferencí, ve kterých tyto týmy odehrály největší počet utkání. Mezi tyto týmy patří: Central Florida, Navy, Notre Dame a Utah State. Pátým nezávislým týmem byl Connecticut, pro který nebyla nalezena komunita žádná. Zajímavý fakt na těchto nezávislých týmech je, že pokud se objeví ve výsledné komunitě pro jakýkoliv tým, tak je z nalezené komunity znát, že se algoritmus během jeho první fáze zaměřil i na sousední komunitu. Mezi tyto týmy patří například Las Vegas - Nevada.

Existuje několik případů, při kterých se zdá, že tento algoritmus selhal. Při detailním pohledu na rozpis utkání však zjistíme, že počet odehraných utkání s týmy z vlastní konference a týmy z konferencí ostatních není vždy uniformní. Podívejme se například na konference, u kterých se podle tabulky 1 zdá, že je algoritmus nepřesný. Nejmenší průměrné hodnoty pro *přesnost* a *úplnost* jsem během testování zaznamenal u konference *Sun Belt* a *Western Athletic*. Při náhledu do rozpisu zápasů jsem zjistil, že je konference *Sun Belt* rozdělena na dvě části a její členové se seskupili právě s týmy konference *Western Athletic*. Tento fakt je způsoben tím, že týmy z *Sun Belt* hrály téměř stejný počet utkání proti týmům z vlastní konference jako proti týmům z konference *Western Athletic*. Navíc 4 ze 7 týmů konference *Sun Belt* vyhodnotilo nezávislý tým *Utah State* jako člena pro jejich nalezenou komunitu.

Je celkem logické, že tento algoritmus selže v situacích, při kterých struktura této sítě neodpovídá přesně složení konferencí. Nicméně ve většině ostatních případů tato me-

toda nalezla komunity, které odpovídaly konferencím. Při 100% úspěšnosti při nalezení správných prvků se bude rovnat *přesnost* , *úplnost* a jejich *harmonický průměr* hodnotě 1. Při analýze tohoto testu jsem došel k hodnotám, které se ve většině případů blíží ideálním (viz tabulka 1). Tato nepatrná chybovost byla způsobena chybami, které jsem uvedl výše. Vizualizaci těchto dat můžeme vidět na obrázku 3.1.2. Pozice jednotlivých týmů jsou v tomto grafu shlukovány do skupin. Každá z těchto skupin odpovídá jedné reálné konferenci. 5 nezávislých týmů je umístěno uprostřed grafu.



Obrázek 3: Síť: Vizualizace sítě reprezentující rozpis zápasů Divize 1-A na sezónu 2000

3.1.2 Porovnání se sezónou 2006

Během šesti let (2000 - 2006) došlo v Divizi 1-A k několika změnám. Do některých konferencí byly přidány nové týmy a z některých byly naopak přesunuty určité týmy do jiných konferencí. Také bylo v rozpisu utkání pro sezónu 2006 navíc 61 týmů z nižších divizí, které ale hrály jen nepatrné množství utkání. Tyto vrcholy byly vyhodnoceny jako vrcholy, které nepatří do žádné konference, takže na výsledek průzkumu stejně neměly vliv. Tyto dvě na první pohled rozdílné sítě lze porovnávat, protože i přes výše uvedené detaily zůstává rozpis utkání na každou sezónu téměř totožný. Vyhodnocování těchto dvou sezón bude provedeno na základě výsledných údajů z tabulky 1. Výsledky experimentu pro sezónu 2006 pochází z článku [3], ve kterém byla metoda používající metriku *L* poprvé představena.

Na první pohled můžeme vidět, že pro tento algoritmus bylo o něco jednodušší nalézt komunity odpovídající cílovým konferencím pro sezónu 2006. Největší rozdíl hodnot *přesnosti*, *úplnosti* a jejich *harmonického průměru* se vyskytl v konferenci *Sun Belt*. Po nahlédnutí do složení konferencí si můžeme povšimnout, že tato konference prošla nejvýznamnějšími změnami ze všech konferencí. Přibylo několik týmů (jmenovitě *Troy, Florida Atlantic,, Florida International*) a naopak tato konference o 2 týmy přišla (*Idaho, New Mexico State*). Tyto změny vyřešily problém popsaný v předešlé kapitole 3.1.1. Po těchto změnách přestaly týmy konference *Sun Belt* hrát většinu mezikonferenčních utkání s jedinou konferencí *Western Athletic* a tím se struktura rozpisu utkání pro tuto konferenci začala podobat všem ostatním konferencím.

Pokud srovnáme průměrné hodnoty pro celou síť z tabulky 1 , můžeme potvrdit, že struktura rozpisu utkání Divize 1-A prošla jistou změnou, ale tato změna nebyla nijak extrémně výrazná. Zůstala z většinové části stejná jako v roce 2000. Nepatrné rozdíly v průměrných hodnotach pro celou divizi byly způsobeny výše uvedenou změnou v konferenci *Sun Belt* a také nepatrnou změnou struktury celé sítě.

3.1.3 Porovnání metriky L a metody využívající lokální modularity

V předešlém experimentu jsem dokázal velkou podobnost struktury grafu pro dvě sezóny Divize 1-A. Na základě tohoto zjištění jsem provedl další experiment obdobný tomu, který byl proveden v kapitole 3.1.2. Jediný rozdíl byl v tom, že pro sezónu 2006 byla tentokrát využita metoda (viz 2.2) využívající k vyhodnocení kvality komunity lokální modularitu(dále jen algoritmus *R*). Cílem tohoto experimentu je dokázat, že metoda využívající metriky *L*, dosahuje přesnějších výsledků při hledání lokálních komunit. I přes to, že toto porovnání nemůže být absolutně přesné kvůli nepatrným rozdílům ve strukturách grafů obou sezón, jsem dokázal, že metoda využívající metriky L dosahuje lepších výsledků než algoritmus *R*. Výsledné hodnoty pro algoritmus *R* byly zveřejněny v článku [3]. Tabulka 2 zobrazuje průměrné hodnoty *úplnost*, *přesnost* a jejich *harmonický průměr*. Označení sloupců této tabulky odpovídá označení sloupců tabulky 1.

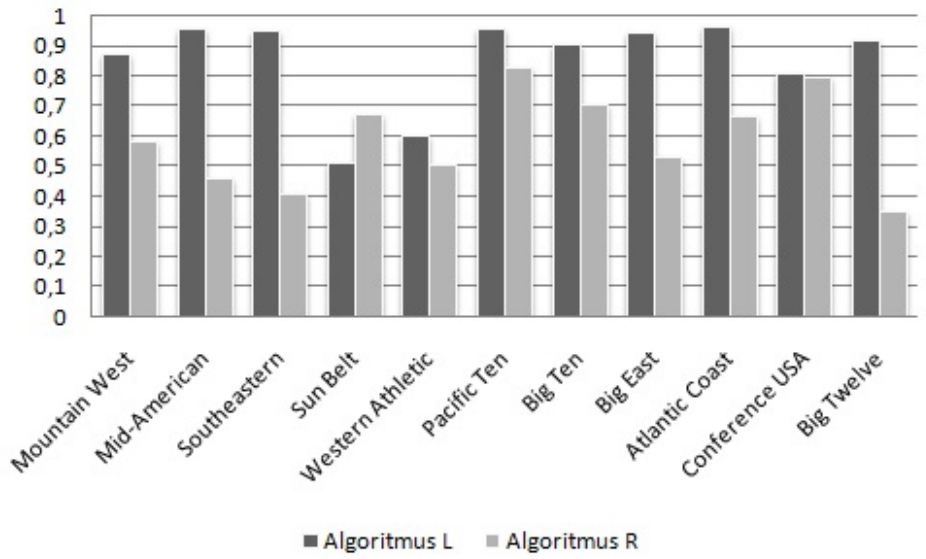
Jako první se při vyhodnocování výsledných hodnot zaměříme na algoritmus *R*. Nízká hodnota *přesnosti* značí velké množství nesprávně určených vrcholů. Toto zjištění potvrzuje problém tohoto algoritmu popsaný v kapitole 2.3.2.

Divize 1-A	**Algoritmus *R***				**Algoritmus *L***				
Konference	**n**	**P**	**R**	**F**	**n**	**X**	**P**	**R**	**F**
Mountain West	9	0,505	0,728	0,588	8	0	0,878	0,875	0,876
Mid-American	12	0,392	0,570	0,461	13	0	0,925	1	0,961
Southheastern	12	0,331	0,541	0,410	12	0	0,924	1	0,952
Sun Belt	8	0.544	0,891	0,675	7	0	0.531	0,51	0,513
Western Athletic	9	0,421	0,716	0,51	10	4	0,648	0,34	0,603
Pacific Ten	10	0,714	1	0,833	10	2	0,932	0,8	0,96
Big Ten	11	0,55	1	0,71	11	9	0,844	1	0,907
Big East	8	0,414	0,781	0,534	8	3	0,914	1	0,945
Atlantic Coast	12	0,524	0,924	0,668	9	2	0,949	1	0,969
Conference USA	12	0,661	1	0,796	10	1	0,822	0,811	0,81
Big Twelve	12	0,317	0,465	0,355	12	0	0,914	0,924	0,922
Celkem	115	0,488	0,783	0,595	110	21	0,844	0,842	0,856

Tabulka 2: Tabulka: Výsledné průměrné hodnoty pro sezónu 2006 dosažené pomocí algoritmu R

U samotného srovnání těchto dvou algoritmů bych jako první připomenul, že algortimus *L* pracoval se sezónou 2000 a algoritmus *R* se sezónou 2006. V experimentu 3.1.2 jsem dokázal, že struktura grafu pro sezónu 2006 obsahuje kvalitnější lokální komunity. Na základě tohoto zjištění můžeme očekávat, že pro algoritmus hledající lokální komunity v grafu pro sezónu 2006, bude jednodušší identifikovat komunitu a přitom dosáhnout větší *přesnosti a úplnosti*. Nicméně i přes tuto nevýhodu dosáhl algoritmus *L* ve většině konferencí lepších hodnot *harmonického průměru*. Jediná konference, ve které dosáhl algoritmus *R* lepší hodnoty *harmonického průměru*, byla konference *Sun Belt*. Ani tento fakt nelze objektivně považovat jako úspěch pro algoritmus *R*. V kapitole 3.1.2 jsem popsal, k jakým změnám během 6-ti let v této konferenci došlo. V grafu na obrázku 4 je znázorněno porovnání hodnot *harmonického průměru*.

Další poznatek, kterého je vhodné si povšimnout je takový, že se algoritmu *L* nepodařilo nalézt lokální komunitu pro 19% vrcholů této sítě. Jednalo se o vrcholy, které se nacházely v okrajových částech příslušných komunit. Postup, jakým algoritmus vyhodnotil, že pro dané vrcholy neexistuje lokální komunita, byl popsán v kapitole 2.4.3. Možným řešením pro tento problém je použití většího počtu počátečních vrcholů.Taková situace může nastat i pro algoritmus *R*, který však nedokáže odhalit chybu a vratí pro tyto vrcholy chybně identifikovanou komunitu. Tento fakt je považován za jeden z hlavních nedostatků algoritmu *R* a je jedním z důvodů, proč tento algoritmus dosahuje tak nízkých hodnot *přesnosti*.



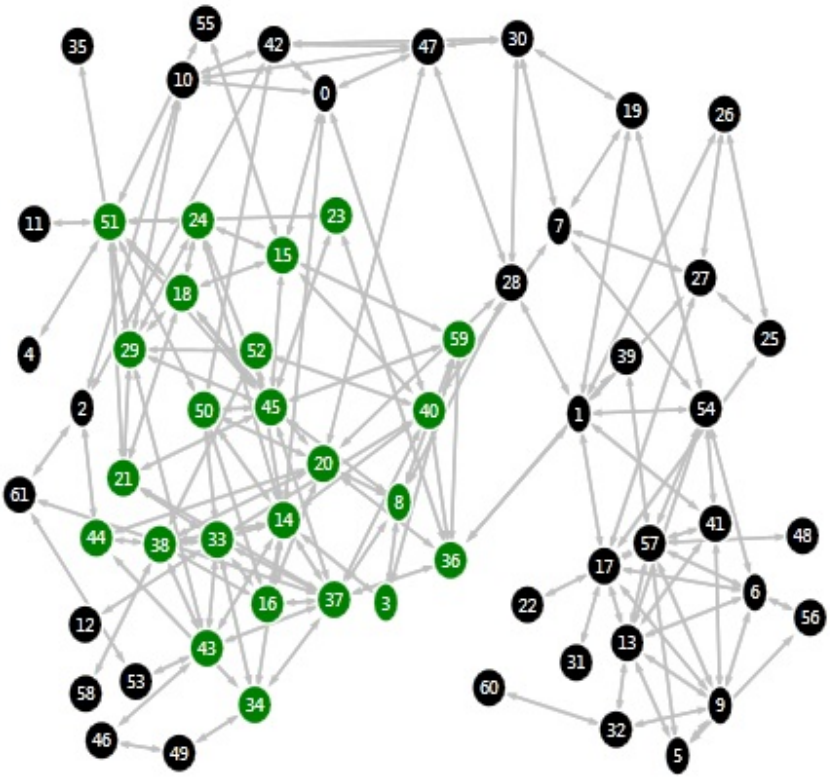
Obrázek 4: Graf: Porovnání hodnot harmonického průměru pro algoritmy R a L

3.2 Nalezení komunit v sociální síti deflínů

Jako data pro další experiment byla použita sociální síť častého sdružování delfínů [4], kteří žíjí poblíž Nového Zélandu. Tato síť je složená z 62 vrcholů. Každý z těchto vrcholů reprezentuje jednoho delfína. Jednotlivé vrcholy mezi sebou spojují hrany, které představují sdružování delfínů mezi sebou. Výsledky nalezených komunit v této síti byly použity pro studii života delfínů, kteří měli tendenci se sdružovat do skupin. V těchto studiích se vyhodnocoval např. věk a pohlaví delfínů v jednotlivých komunitách (skupinách).

Během experimentu byl použit každý z 62 vrcholů jako počáteční n_0 pro algoritmus využívající metriky L. U 13-ti případů (21%) bylo vyhodnoceno, že pro ně nebyla nalezena komunita, pokud byly použity jako počáteční vrchol. Pouze 8 z nich nebylo členy žádné z nalezených komunit. Jedná se o vrcholy 1, 12, 17, 22, 31, 39, 48 a 58. Většinou se jedná buď o vrcholy typu hub (1, 17) nebo vrcholy, které jsou velmi špatně spojeny se zbytkem sítě (22, 31, 48).

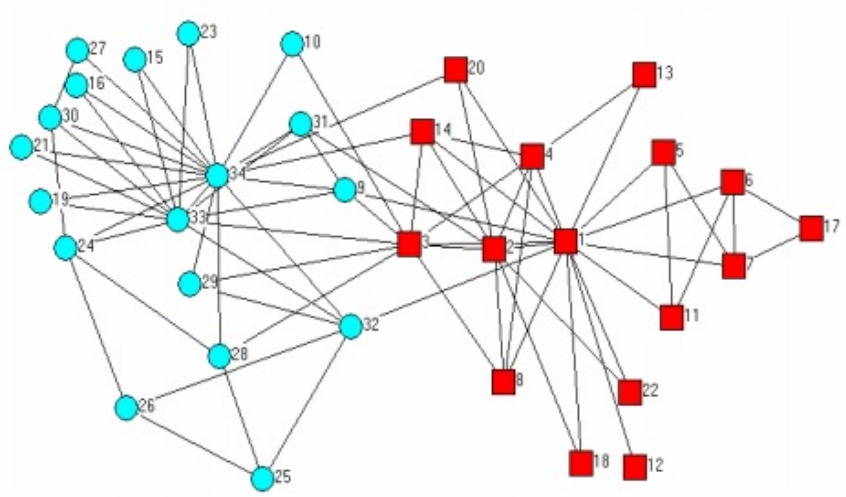
Při tomto experimentu lze dobře ukázat rozdíl mezi výsledky globálních a lokálních metod pro vyhledávání komunit. Globální algoritmus by rozdělil celý graf na části a při tom pokryl všechny vrcholy. Lokální metoda se však zastaví po dosažení určité hodnoty koeficientu kvality, pokud už neexistuje možnost, jak tuto komunitu vylepšit. Toto vysvětluje vysoký počet nalezených komunit pro lokální metodu (přibližně 13 - komunity lišící se pouze v jednom vrcholu byly pro přehlednost započítány jako jedna). Ukázka vizualizace této sítě je zobrazena na obrázku 5.



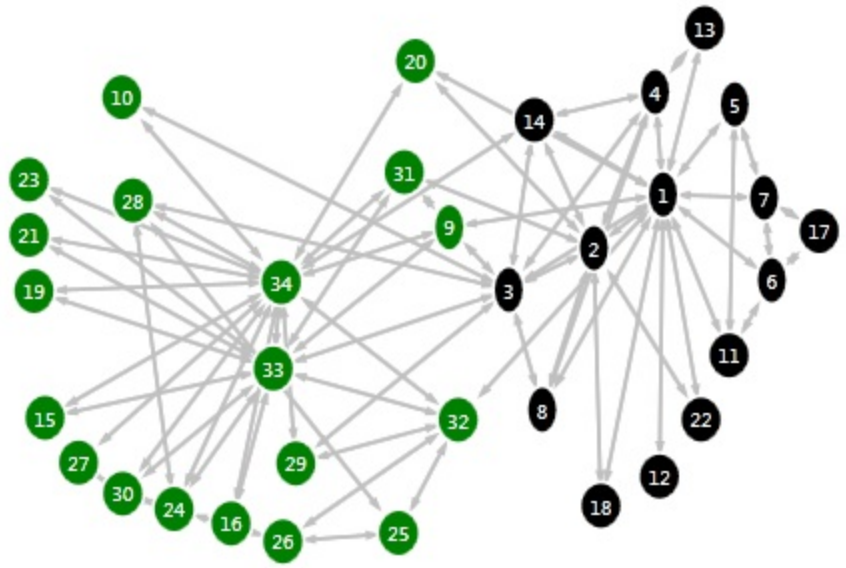
Obrázek 5: Síť: Ukázka druhé největší nalezené komunity v síti delfínů

3.3 Zacharyho karate klub

Tato sociální síť se skládá z 34 vrcholů reprezentujících jednotlivé členy karate klubu jedné americké univerzity. W. W. Zachary [5] studoval během sedmdésátých let minulého století interakci mezi členy tohoto klubu. Jako interakce se v této síti považuje přátelství mezi jednotlivými členy, společné pořádání akcí nebo navštěvování stejné lekce karate kurzu. Během určitého období došlo v klubu k rozepři mezi jeho ředitelem a jedním z instruktorů. Tato rozepře vedla k rozdělení klubu do dvou částí. Instruktor založil nový klub, do kterého přešla přibližně polovina členů z původního. Díky této skutečnosti se Zacharyho karate klub často využívá k testování různých algoritmů pro vyhledávání komunit v grafu.



Obrázek 6: Síť: Ukázka rozdělení původního karate klubu do dvou částí.



Obrázek 7: Síť: Ukázka nalezené komunity pro vrchol 34

Při stejném postupu jako v předchozích experimentech nenalezl tento algoritmus žádnou komunitu pro 6 vrcholů (17%). Pro ostatní vrcholy bylo nalezeno 12 rozdílných komunit. Dosáhl průměrného koeficientu kvality $L = 1.15$

Na obrázku 6 je zobrazeno rozdělení na dvě komunity, které odpovídají jednotlivým částem po rozpadu klubu. Vrcholy kulatého tvaru představují skupinu okolo instruktora. Čtvercové představují skupinu okolo ředitele klubu. Na druhém obrázku 7 je komunita nalezená algoritmem pro vyhledávání lokálních komunit za použití metriky L. Takovou komunitu nalezl tento algoritmus pro dva případy počátečních vrcholů, a to pro vrchol 27 a 34. Pokud porovnáme obrázky 6 a 7, můžeme říct, že nalezená komunita odpovídá jedné z částí, na které byl klub rozdělen. Jediný nesprávně vyhodnocený vrchol byl v tomto případě vrchol 20. Úspěšnost algoritmu nalézt komunitu pro jednu z částí, můžeme považovat za jeden z dalších důkazů o jeho přesných výsledcích.

4 Testovací aplikace

Tato kapitola se bude věnovat aplikaci, kterou jsem navrhl a naimplementoval. Tato aplikace mi posloužila pro vizualizaci sítí během provádění experimentů.

4.1 Technologie

Jako hlavní technologie byl použit WPF (Windows presentation foundation). Dále jsem se rozhodl pro využití frameworku nazvaného *GraphSharp* [9]. Jedná se o framework pro vykreslování různých druhů sítí. Jako hlavní výhodu *GraphSharpu* jsem viděl možnost použití algoritmů pro odstranění překrývání hran při zobrazování grafů. Konkrétně je v aplikaci využit algoritmus na odstranění překrývání hran - *Kamada - Kawai*.

4.2 Uživatelské rozhraní - ovládání

Návrh grafického uživatelského rozhraní jsem přizpůsobil účelu, pro který měla být tato aplikace navržena - Aplikační prostředí pro experimenty. To byl také jeden z důvodů, proč jsem použil jednoduché a intuitivní ovládání. Pokud chceme nalézt lokální komunitu pro určitý vrchol, zvolíme ho tzv. dvojklikem. Tím spustíme algoritmus využívající metody L s počátečním vrcholem, který byl vybrán dvojklikem.

Výsledný vzhled vykreslených grafů můžeme vidět na většině obrázků v této práci (např. 3, 7).

4.3 Data

Tato aplikace obsahuje několik sítí, na kterých byly prováděny experimenty. Následující seznam uvádí jména jednotlivých sítí podle toho, jak jsou pojmenovány v aplikaci.

- Karate - Zacharyho Karate Klub - [5]
- Dolphins - Síť komunity delfínů - [4]
- Football - Síť vysokoškolské ligy amerického fotbalu - [6]
- Adjnoun - [7]
- polbooks - [8]
- example1, example 2 - jedná se o grafy, na kterých jsem testoval správnou implementaci algoritmu L

5 Závěr

Cílem mé práce bylo provedení průzkumu metod pro vyhledávání lokálních komunit v sociálních sítích. Tento průzkum jsem provedl ve dvou fázích.

V první fázi jsem se zaměřil na teoretický průzkum těchto metod. U některých z nich jsem poukázal na jejich nedostatky a na praktických triviálních příkladech jsem tyto nedostatky předvedl. U některých metod jsem zmínil, jakým způsobem zdokonalily řešení problémů, které se vyskytly u předchozích algoritmů.

Ve druhé fázi průzkumu jsem provedl několik testů na reálných sociálních sítích. K těmto testům jsem vybral a naimplementoval metodu L, která byla v první fázi průzkumu vyhodnocena jako metoda, která dosahuje nejlepších výsledků. Podle literatury tato metoda zvládla řešit problémy, se kterými si ostatní zmíněné metody nevěděly rady. Během testování jsem došel k několika poznatkům. Tato metoda opravdu dokázala rozpoznat nejen zajímavé lokální komunity, ale také tyto komunity identifikovala s velmi vysokou přesností ve srovnání s reálnými komunitami. Provedl jsem také experiment, při kterém jsem srovnal přesnost a úplnost nalezených komunit mezi metodou L a metodou R. Během tohoto experimentu jsem dokázal nedostatky metody R, na které jsem poukázal v první fázi průzkumu.

Jako závěr tohoto vyhodnocení bych rád uvedl, že se naplnily předpoklady o kvalitě metody L. Komunity nalezené pomocí této metody dosahovaly téměř ideálních hodnot přesnosti a úplnosti. Během srovnání s ostatními algoritmy tato metoda nalezla ve všech případech mnohem přesnější výsledné komunity.

Navrhl jsem také jednoduchou aplikaci pro vizualizaci sítí. Aplikace je schopná interaktivně vyhledávat komunity pro uživatelem zvolený vrchol. Tato aplikace využívá výše zmíněnou metodu L.

Pokud se zamyslím, jakým přínosem by mohla být tato práce do budoucnosti, tak mně jako nejvhodnější způsob využití napadá testování novějších a kvalitnějších metod. Metod, které by například odstranily problém metriky L s nenalezením komunity v některých případech při použití počátečního vrcholu z okrajové části grafu. Výsledky experimentů v této práci by mohly být použity pro srovnání s nalezenými komunitami pomocí nových algoritmů.

Jan Freiherr

6 Reference

[1] CLAUSET, Aaron, *Finding local community structure in networks.* [online], [cit. 2013-04-07]. Physical Review E, vol.72, p. 026132,2005, Dostupné z: http://www.cs.unm.edu/ moore/tr/05-02/local_communities.pdf

[2] F. LUO, J. Z. WANG, PROMISLOW E., *Exploring local community structures in large networks* [online], [cit. 2013-04-15]. Dostupné z: http://dl.acm.org/citation.cfm?id=1249100

[3] CHEN, Jiyang, Osmar R. ZAIANE a Randy GOEBEL, *Local Community Identification in Social Networks.*[online] University of Alberta, Canada.[cit. 2013-04-08] Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.158.5574

[4] D. LUSSEAU, K. SCHNEIDER, O.J BOISSEAU, HAASE P., SLOOTEN E. a M. S. DAWSON *The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations*[cit. 2013-04-20].Behavioral Ecology and Sociobiology 54, 396-405 (2003)

[5] W. W. ZACHARY *An information flow model for conflict and fission in small groups*[cit. 2013-04-20] Journal of Anthropological Research 33, 452-473 (1977).

[6] M. GIRVAN, M. E. J. NEWMAN *Compiled data for fall season of Division 1 - A*[online]Květen 2013,Proc. Natl. Acad. Sci. USA 99, 7821-7826 (2002) Dostupné z: http://www-personal.umich.edu/m̃ejn/netdata/football.zip

[7] M. E. J. NEWMAN *Compiled data for word adjacencies* květen 2013, Phys. Rev. E 74, 036104 (2006). Dostupné z: http://www-personal.umich.edu/ mejn/netdata/adjnoun.zip

[8] V. KREBS *Compiled data for network of books of US Politics* květen 2013 Dostupné z: http://www-personal.umich.edu/ mejn/netdata/polbooks.zip

[9] *Graphsharp* - květen 2013. Dostupné z: http://graphsharp.codeplex.com